UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/850,183	05/07/2001	Mark A. Kampe	80168-0106	9753

32658 7590 10/03/2005

HOGAN & HARTSON LLP
ONE TABOR CENTER, SUITE 1500
1200 SEVENTEEN ST.
DENVER, CO 80202

EXAMINER	
PROCTOR, JASON SCOTT	
ART UNIT	PAPER NUMBER
2123	

DATE MAILED: 10/03/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/850,183	Applicant(s) KAMPE, MARK A.
	Examiner Jason Proctor	Art Unit 2123

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 July 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16,18 and 19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16,18 and 19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 February 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Claims 1-16 and 18-19 are pending in the application. Claims 1, 6, 8 and 16 have been amended. Claims 1-16 and 18-19 have been rejected.

Claim Rejections - 35 USC § 101

35 U.S.C. § 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

1. Claims 1-7 are rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter. MPEP 2106 reads as follows:

> Claims to computer-related inventions that are clearly nonstatutory fall into the same general categories as nonstatutory claims in other arts, namely natural phenomena such as magnetism, and abstract ideas or laws of nature which constitute "descriptive material." Abstract ideas, *Warmerdam*, 33 F.3d at 1360, 31 USPQ2d at 1759, or the mere manipulation of abstract ideas, *Schrader*, 22 F.3d at 292-93, 30 USPQ2d at 1457-58, are not patentable. Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." In this context, "functional descriptive material" consists of data structures and computer programs which impart functionality when employed as a computer component. (The definition of "data structure" is "a physical or logical relationship among data elements, designed to support specific data manipulation functions." The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993).) "Nonfunctional descriptive material" includes but is not limited to music, literary works and a compilation or mere arrangement of data.

Claims 1-7, as amended, recite a computer readable medium containing a "data structure". However, the recited "data structure" does not satisfy the IEEE definition. Specifically, the claimed "data structure" provides no functionality except for "storing", which is not a specific data manipulation functions. A computer readable medium is capable of storing data and does not require the claimed "data structure". In contrast, a specific data manipulation function could be, for example, specific binary tree algorithms for manipulating a binary tree.

Claims 1-7 recite nonfunctional descriptive material stored on a computer readable medium and are therefore nonstatutory.

To expedite a complete examination of the instant application the claims rejected under 35 U.S.C. § 101 (nonstatutory) above are further rejected as set forth below in anticipation of applicant amending these claims to place them within the four statutory categories of invention.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-2 are rejected under 35 U.S.C. § 103(a) as being unpatentable over US Patent No. 5,014,220 to McMann et al. (McMann) in view of "Understanding Fault-Tolerant Distributed Systems" by Flaviu Cristian (Cristian) (cited on PTO-892 paper number 20040823).

McMann teaches a system and method for generating a reliability model for a complex system having different classes of failures ["*The present invention provides a reliability model for use by a reliability analysis tool.*" (column 5, lines 37-54); "*A system in SURE is defined as a state space description: the set of all feasible states of the system, given an initial state. State transitions, in SURE, describe the <u>occurrence of faults and fault recovery actions</u> that cause the system to change from one state to another,*" (column 6, lines 3-8, emphasis added); "*For highly*

reliable system, additional functions are incorporated into the architecture for failure detection, isolation, and recovery (FDIR)," (column 5, lines 54-56, emphasis added)];

The reliability model includes a model for defining expected failure rates and time to recover from the expected failures for components of the platform ["*Given the state space description, including an identification of the initial state and those states that represent an unreliable system, SURE computes the upper and lower bounds on system reliability and provides an enumeration of all system failures,*" (column 6, lines 8-12, emphasis added); transitions include recovery actions, as in "*The failure modes and FDIR attributes are described to ASSIST as transitions in the form of logical statements,*" (column 6, lines 20-21, emphasis added); failures and recoveries, represented by transitions, are time and rate dependent, as in "*Another critical aspect of FMEA is concerned with the effects of multiple failures on the system and the effects of nearly simultaneous failures – a particular state of vulnerability in which a second failure may occur before the system can recover from the first failure. These time dependencies contribute to the difficulty of an accurate reliability analysis*", (column 5, lines 58-64)];

A reliability model within the platform reliability model, including an aggregated failure rate for each class of failures and an aggregated repair time for each class of failures for at least one component ["*To manage the analysis complexity, a system may be divided into sets of components.* [...] *This component then becomes a lowest level component in a new aggregate model that also accounts for dependencies among the sets,*" (column 7, lines 20-30, emphasis added)].

McMann does not expressly teach the specific failure modes recited in the claim, however McMann does explicitly teach that the disclosed invention is a framework for developing "a model for a system of virtually any complexity" (column 2, lines 11-14). McMann further teaches the suitability of the reliability modeling system and method to other disciplines [*"These units may correspond to a physical hardware device or may refer to assemblies of units for which composite failure modes are identified. The units have been referred to in literature by various nomenclature including systems and subsystems, assemblies and subassemblies, components and subcomponents, structures and substructures, etc."* (column 6, lines 56-63)]. McMann also teaches a computer hardware example (column 7, lines 4-19)].

Cristian teaches the four failure modes recited in the claim as known in the art:

"failures that can be corrected internally with no loss of service" [*"A* timing *failure occurs when the server's response is functionally correct but untimely – the response occurs outside the real-time interval specified,"* (page 58, center and right columns)],

"failures that can be corrected by a restart with no loss of state" [*"If, after a first omission to produce output, <u>a server omits to produce output to subsequent inputs until its restart</u>, the server is said to suffer a* crash *failure"*; and *"A pause-crash occurs when a server restarts in the state it had before the crash,"* (page 58, right column, emphasis added)],

"failures that can be corrected by a restart with loss of state" [*"If, after a first omission to produce output, <u>a server omits to produce output to subsequent inputs until its restart</u>, the server is said to suffer a* crash *failure"*; and *"An amnesia-crash occurs when the server restarts in a predefined initial state that does not depend on the inputs seen before the crash,"* (page 58, right column, emphasis added)],

"failures that can be corrected by fail over" [*"A halting-crash occurs when a crashed server never restarts,"* (page 58, right column, emphasis added); Official notice is taken that a person of ordinary skill in the art would recognize that a "fail over" is the necessary recovery action from a "halting-crash"].

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the failure modes taught by Cristian for the purposes of modeling the availability of software and hardware systems with the reliability model generation system of McMann to arrive at the claimed invention; an availability model for software with the particular failure modes. Although much of McMann is directed toward a reliability model, McMann also provides sufficient teaching of recovery actions to suggest adapting the invention into an availability model, such an adaptation being described as known in the art by Malek. Motivation to combine the failure modes taught by Cristian with the reliability model generation of McMann would be found in the nature of the problem to be solved [*"The task of designing and understanding fault-tolerant distributed system architectures is notoriously difficult"*, (Cristian, page 57, left column); which complements *"The use of digital systems and redundancy management schemes to satisfy flight control system requirements of high performance aircraft has increased both the number of implementation alternatives and the overall system design complexity. Consequently, a comprehensive reliability analysis of each candidate architecture becomes tedious, time-consuming, and costly"*, (McMann, column 1, lines 29-35)], that is, designing and understanding a complex system.

In response, Applicants argue primarily that:

> McMann provides no teaching of modeling software as a component in its reliability models of networks or computer systems. The lack of teaching of modeling software as a component in a network or system can be seen in a review of McMann with reference to Figure 2, which shows that the input for the model builder 202 is provided by knowledge bases 214 and 216. Reading McMann from col. 8, line 17 to col. 9, line 43, it can be seen that components described in detail are only hardware components and no discussion of software applications and their failure modes or recovery is provided by McMann. Further, McMann in Figure 4A shows "a typical computer system which may be represented in the BBD 214" (see, McMann beginning at col. 9, line 44).

The Examiner respectfully traverses this rejection as follows.

The Examiner has reviewed the portions of McMann cited by Applicants (Figure 2; column 8, line 17 – column 9, line 43) but does not find clear basis for Applicants' conclusion. In particular, McMann teaches that *"the inputs 102 consist of two knowledge bases 214 and 216 which provide a specification of the functional and structural characteristics of the system"* (column 8, lines 25-29). It appears that Applicants and the Examiner agree that "structural characteristics" may comprise hardware components. The previous rejection pointed out McMann's computer hardware example, also noted by Applicants. However, the Examiner respectfully submits that a person of ordinary skill in the art would understand "functional characteristics" of a computer system as including, among other constructs, the software components of that computer system, which software components define the functional behavior of the system.

While Applicants may be correct that McMann does not anticipate the claimed combination of software and hardware availability models, the Examiner reasserts that, to a person of ordinary skill in the art and in combination with the Cristian reference, it would have been obvious to incorporate "software components" as defining the "functional characteristics" of a computer system as suggested by McMann.

The Examiner respectfully submits that Applicants' arguments regarding the specific language of claim 1 have been addressed above.

Applicants' reiterate the allegation that McMann does not teach modeling software components in the context of several claims, all of which would be addressed as above.

Regarding the Cristian reference, Applicants' argue primarily that:

> Cristian as cited in the Office Action is only discussing hardware failure modes (such as server failure). Hence, the combined teaching of these two references fails to discuss modeling of software components in a network.

The Examiner respectfully traverses this argument as follows.

The Examiner concurs that Cristian discusses servers. However, the Examiner respectfully disagrees with Applicants' conclusion that Cristian is only discussing hardware failure modes. See Cristian, page 57, right column:

> Servers can be hardware or software implemented. For example, a 4381 raw processor service is typically implemented by a hardware server; however, sometimes one can see this service "emulated" by software. A DB2 service is typically implemented by software, although it is conceivable to implement this service by a hardware database machine.

Cristian's use of the term appears to be standard in the art. The following is excerpted from the definition of "server" provided by IEEE 100: The Authoritative Dictionary of IEEE Standards Definitions, Seventh Edition:

> **server** (1) A system component that performs operations required for the processing of a call.
> **server** (3) In a network, a device or computer system that is dedicated to providing specific facilities to other devices attached to the network.
> **server** (4) The facility in the terminal or work station that provides input (keyboard, mouse) and output (screen graphics) services to the application.
> **server** (5) The software component on one device that provides services for use by clients on the same or another device.

Thus it appears to the Examiner that the art-recognized definition of "server" refers to both hardware and software, in agreement with Cristian's explicit disclosure of the same. Applicants' allegation that Cristian is only discussing hardware failure modes is unpersuasive.

Additionally, the Examiner respectfully submits that none of the recited limitations in claim 1 involve "software components in a network" either explicitly or implicitly. Applicants' arguments appear to be narrower in scope than the claim.

Applicants further argue that:

> The Examiner cites McMann for teaching the concept of placing failures into classes but at the citation McMann discusses "sets of components" but not sets of or classes of failures for its hardware components. [...] For this additional reason, McMann fails to teach each and every limitation of claim 1.

The Examiner respectfully traverses this argument as follows.

At the portion cited (column 7, lines 20-30), McMann discloses that "critical failures modes are ascertained" from "the components in each set". Therefore McMann does disclose "classes of failures", specifically the failures that correspond to the different classes or sets of components. Further, the Examiner is unaware of a requirement that a single reference teach each and every limitation of a claim in a rejection under 35 U.S.C. § 103.

Applicants' arguments have been fully considered but have been found unpersuasive.

Regarding claim 2, McMann teaches platform parameters define platform problems causing failures ["*Failure modes of subcomponents are combined according to their severity and common effects on a higher level component. <u>These failure modes are used to define a model of the component at the higher level.</u> This component then becomes a lowest level component in a new aggregate model that also accounts for dependencies among the sets",* (column 7, lines 24-30, emphasis added)] and affecting recovery times related to the platform problems ["*This component then becomes a lowest level component in <u>a new aggregate model that also accounts</u>*

<u>*for dependencies among the sets*</u>*",* (column 7, lines 28-30, emphasis added); *"Therefore, to determine the sequence of component failures that contribute to a particular undesirable condition, a Failure Mode Effect Analysis (FMEA) is performed that traces* <u>*the effects of component failures according to component interactions*</u>*. For highly reliable systems, additional functions are incorporated into the architecture for* <u>*failure detection, isolation, and recovery (FDIR)*</u>*",* (column 5, lines 49-56, emphasis added)] and wherein at least a portion of the platform parameters are used to determine the aggregated repair time [*"Once a* <u>*top level reliability model 210*</u> *is defined, further reduction techniques are applied by the model reducer/encoder 204 of FIG. 2,* <u>*to reduce the model state space and encode the global model into the ASSIST syntax from which the SURE model is built*</u>*",* (column 9, lines 18-23, emphasis added); *"*<u>*A system in SURE is defined as a state space description*</u>*: the set of all feasible states of the system, given an initial state.* <u>*State transitions*</u>*, in SURE,* <u>*describe the occurrence of faults and fault recovery actions*</u> *that cause the system to change from one state to another",* (column 6, lines 3-8, emphasis added)].

In response, Applicants argue primarily that:

> There is no teaching in McMann that the platform parameters <u>define platform problems causing failures and affecting recovery times related to the platform problems.</u> Further, McMann does not teach that "at least a portion of the platform parameters are used to determine the aggregated repair time" that would include repair of the at least one software component with McMann not teaching the determination of such an aggregate repair time including repair of a software component nor using platform parameters to determine it.

The Examiner respectfully traverses this argument as follows.

The Examiner respectfully submits that the portions of McMann cited in the rejection address Applicants' argument. The Examiner's interpretation of the claims, which has not been

challenged, is clear from the body of the rejection. McMann teaches "platform parameters" (a model of the component at a higher level) and "affecting recovery times related to the platform problems" (a new aggregate model that accounts for dependencies among the sets). Applicants' second point appears to depend on McMann's alleged deficiency regarding software components, which have been addressed above. If that interpretation is incorrect, the Examiner apologizes and respectfully requests clarification of the argument.

Applicants arguments have been fully considered but have been found unpersuasive.

3. Claim 3 is rejected under 35 U.S.C. § 103(a) as being unpatentable over McMann in view of Cristian as applied to claim 1 above, and further in view of US Patent No. 4,870,474 to Rutenberg.

McMann teaches the capability to include a hardware component availability model within the platform availability model ["*To manage the analysis complexity, a system may be divided into sets of components*", (column 7, lines 20-21)] and provides a multiprocessor example (column 7, lines 4-19). While McMann teaches the capability to include a hardware component availability model, such a model is not explicitly disclosed.

Cristian teaches various faults, including hardware faults, but does not explicitly disclose combining a hardware component availability model within a platform availability model.

Rutenberg teaches a fault-tree analysis which can detect all latent hardware and software design defects that could cause unanticipated critical failure of a complex software controlled electronic system (abstract). Rutenberg explicitly teaches motivation for performing a combined

hardware and software fault analysis ["*As discussed above, a complete analysis of the critical failure potential of a design can only result from an understanding of all the possible interactions between the system hardware and its control software*", (column 6, lines 7-11)].

In light of the Rutenberg teachings and motivation, it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to include a hardware component availability model when using the system and method of generating a reliability model taught by McMann. Such a hardware component availability model would be yet another set of components of the larger system, as taught by McMann. The motivation for making the combination could be as explicitly taught by Rutenberg, that is, to perform a complete analysis of the interactions between the hardware and software of a system.

4. Claim 4 is rejected under 35 U.S.C. § 103(a) as being unpatentable over McMann in view of Cristian as applied to claim 1 above, and further in view of "Survey of Software Tools for Evaluating Reliability, Availability, and Serviceability" by Allen M. Johnson, Jr., and Mirslaw Malek (Malek).

McMann does not explicitly teach that the aggregated repair time includes a time to detect and identify an error associated with running the at least one software component on said platform.

Cristian does not explicitly teach that time to detect and identify an error contributes to an aggregated repair time.

Malek teaches numerous concepts known in the art related to availability and reliability modeling of computing systems and software (section 1.2, "Service Cost and Repair Time Model"), in particular a model for mean time to repair (MTTR) (page 223, left column). This model includes "a time to detect and identify an error", broken into several components such as $T_A(i)$, "average time to talk to customer and obtain the fault symptoms, identification of failing unit, and any other preparation required in the *i*th month"; T_B, "time required to run diagnostics or analyze information logged at the time of the error to determine the fault symptoms"; U_{diag}, "application factor to obtain the additional time required when the diagnostics or logout analysis are not effective in isolating the problem to a single RU" (*replaceable units*, page 232, right column); and P_{isol}, "probability that the error symptoms uniquely identify the failing RU (depending upon the maintenance strategy applied)". The goal of Malek's MTTR model is to accurately calculate the hours spent servicing a given type of system (page 232, right column).

It would have been obvious to a person of ordinary skill in the art to combine the accurate MTTR model taught by Malek with the reliability model generation system and method taught by McMann to achieve a more accurate model of the recovery actions in the system. Such a combination could be achieved by incorporating the MTTR model calculations taught by Malek into the state transitions of the SURE model generated by McMann's system and method. Motivation to combine would be apparent to a person of ordinary skill in the art as a result of the accuracy and comprehensiveness of Malek's MTTR model.

5. Claim 5 is rejected under 35 U.S.C. § 103(a) as being unpatentable over McMann in view of Cristian as applied to claim 1 above, and further in view of "Availability analysis of a certain class of distributed computer systems under the influence of hardware and software faults" by G.D. Hassapis (Hassapis)

McMann teaches the capability to generate a reliability model where the system is a node in a network [*"Briefly described, the present invention contemplates a reliability model generator which automatically generates a composite reliability model for a system of virtually any complexity"*, (column 2, lines 11-14); *"To manage the analysis complexity, a system may be divided into sets of components*", (column 7, lines 20-21)]. While McMann teaches the capability to generate a reliability model for node in a network, such a model is not explicitly disclosed.

Cristian teaches various faults, including server faults, but does not explicitly disclose a reliability model including a node in a network.

Hassapis teaches an availability model for a computer platform with at least one software component [*"...assess the availability when the system is subjected to the combined effects of hardware and software faults either during its normal operating time or repair time."* (abstract)] wherein the hardware platform is a node in a network [*"This theory has been made more appropriate for the type of <u>software used in the distributed process control systems</u> and has been extended by incorporating the state of the computer hardware at time* t *explicitly*", (page 524, right column, emphasis added); *network processor, network interface,* etc., page 527, Fig. 1].

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Hassapis, regarding a combined hardware

software availability analysis wherein the hardware is a node in a network, with the reliability model generation system and method of McMann in order to accurately assess the availability of a complex system such as a distributed computing system. The combination could be achieved representing a node in a network as the system of components described by McMann. Motivation to do so would be found in the nature of the problem to be solved, such as the need to analyze the availability of a node in a network, wherein the node comprises both hardware and software.

6. Claim 6 is rejected under 35 U.S.C. § 103(a) as being unpatentable over McMann in view of Hassapis.

McMann teaches a system and method for generating a reliability model for a complex system having different classes of failures ["*The present invention provides a reliability model for use by a reliability analysis tool.*" (column 5, lines 37-54); "*A system in SURE is defined as a state space description: the set of all feasible states of the system, given an initial state. State transitions, in SURE, describe the* <u>*occurrence of faults and fault recovery actions*</u> *that cause the system to change from one state to another,*" (column 6, lines 3-8, emphasis added); "*For highly reliable system, additional functions are incorporated into the architecture for* <u>*failure detection, isolation, and recovery (FDIR),*</u>" (column 5, lines 54-56, emphasis added)];

The reliability model includes a model for defining expected failure rates and time to recover from the expected failures for components of the platform ["*Given the state space description, including an identification of the initial state and those states that represent an*

unreliable system, SURE computes the <u>upper and lower bounds on system reliability</u> and provides an enumeration of all system failures," (column 6, lines 8-12, emphasis added); transitions include recovery actions, as in "*The failure modes and <u>FDIR attributes</u> are described to ASSIST as <u>transitions in the form of logical statements</u>,*" (column 6, lines 20-21, emphasis added); failures and recoveries, represented by transitions, are time and rate dependent, as in "*Another critical aspect of FMEA is concerned with the effects of multiple failures on the system and the effects of nearly simultaneous failures – a particular state of vulnerability in which a second failure may occur before the system can recover from the first failure. These time dependencies contribute to the difficulty of an accurate reliability analysis*", (column 5, lines 58-64)].

Official Notice is taken that in the case of a network or distributed computing system, it is known in the art that node reboot time significantly contributes to the recovery time of the system and should therefore be contemplated as parameters of the corresponding recovery actions.

A reliability model within the platform reliability model, including an aggregated failure rate for each class of failures and an aggregated repair time for each class of failures for at least one component ["*To manage the analysis complexity, a system may be divided into sets of components.* [...] *<u>This component then becomes a lowest level component in a new aggregate model</u> that also accounts for dependencies among the sets,*" (column 7, lines 20-30, emphasis added)].

McMann teaches the capability to generate a reliability model where the system is a node in a network ["*Briefly described, the present invention contemplates a reliability model*

generator which automatically generates a composite reliability model for a system of virtually any complexity", (column 2, lines 11-14); "*To manage the analysis complexity, a system may be divided into sets of components*", (column 7, lines 20-21)]. While McMann teaches the capability to generate a reliability model for node in a network, such a model is not explicitly disclosed.

Hassapis teaches an availability model for a computer platform with at least one software component [*"...assess the availability when the system is subjected to the combined effects of hardware and software faults either during its normal operating time or repair time."* (abstract)] wherein the hardware platform is a node in a network ["*This theory has been made more appropriate for the type of <u>software used in the distributed process control systems</u> and has been extended by incorporating the state of the computer hardware at time* t *explicitly*", (page 524, right column, emphasis added); *network processor, network interface, etc.*, page 527, Fig. 1].

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Hassapis, regarding a combined hardware software availability analysis wherein the hardware is a node in a network, with the reliability model generation system and method of McMann in order to accurately assess the availability of a complex system such as a distributed computing system. The combination could be achieved representing a node in a network as the system of components described by McMann. Motivation to do so would be found in the nature of the problem to be solved, such as the need to analyze the availability of a node in a network, wherein the node comprises both hardware and software.

In response, Applicants argue primarily that:

> Hassapis fails to teach in its availability analysis on page 525 a software availability model that includes "an aggregated failure rate" for each software component on a node. Further Hassapis does not teach that its analysis uses "aggregated repair time" for each software component on a node but instead shows use in a stochastic process "0" if a computer module is under repair at a particular time and "1" if the computer module is functioning at that time.

The Examiner respectfully traverses this argument as follows.

Hassapis has not been cited as teaching a software availability model that includes "an aggregated failure rate". McMann has been cited as teaching the limitations related to "aggregated" rates.

Applicants' arguments have been fully considered but have been found unpersuasive.

Regarding claim 7, McMann teaches the capability to include a hardware component availability model within the platform availability model ["*To manage the analysis complexity, a system may be divided into sets of components*", (column 7, lines 20-21)] and provides a multiprocessor example (column 7, lines 4-19). The combination formed in the rejection of claim 6 involves representing a node in a network as a system or set of components in the reliability model generation system and method of McMann. As a node in a network is a hardware component, that combination also teaches the limitations of claim 7.

Claims 8 and 18 are rejected under 35 U.S.C. § 103(a) as being unpatentable over McMann in view of Cristian.

McMann teaches a system and method for generating a reliability model for a complex system having different classes of failures ["*The present invention provides a reliability model for use by a reliability analysis tool.*" (column 5, lines 37-54); "*A system in SURE is defined as a state space description: the set of all feasible states of the system, given an initial state. State transitions, in SURE, describe the occurrence of faults and fault recovery actions that cause the system to change from one state to another,*" (column 6, lines 3-8, emphasis added); "*For highly reliable system, additional functions are incorporated into the architecture for failure detection, isolation, and recovery (FDIR),*" (column 5, lines 54-56, emphasis added)];

The reliability model includes a model for defining expected failure rates and time to recover from the expected failures for components of the platform ["*Given the state space description, including an identification of the initial state and those states that represent an unreliable system, SURE computes the upper and lower bounds on system reliability and provides an enumeration of all system failures,*" (column 6, lines 8-12, emphasis added); transitions include recovery actions, as in "*The failure modes and FDIR attributes are described to ASSIST as transitions in the form of logical statements,*" (column 6, lines 20-21, emphasis added); failures and recoveries, represented by transitions, are time and rate dependent, as in "*Another critical aspect of FMEA is concerned with the effects of multiple failures on the system and the effects of nearly simultaneous failures – a particular state of vulnerability in which a second failure may occur before the system can recover from the first failure. These time dependencies contribute to the difficulty of an accurate reliability analysis*", (column 5, lines 58-64)];

Failure rates and recovery rates are used to generate state transition parameters ("*State transitions, in SURE, describe the occurrence of faults and fault recovery actions that cause the system to change from one state to another*", (column 6, lines 5-8)];

A reliability model within the platform reliability model, including an aggregated failure rate for each class of failures and an aggregated repair time for each class of failures for at least one component ["*To manage the analysis complexity, a system may be divided into sets of components.* [...] *This component then becomes a lowest level component in a new aggregate model* *that also accounts for dependencies among the sets,*" (column 7, lines 20-30, emphasis added)].

McMann does not expressly teach the specific errors recited in the claim, however McMann does explicitly teach that the disclosed invention is a framework for developing "a model for a system of virtually any complexity" (column 2, lines 11-14). McMann further teaches the suitability of the reliability modeling system and method to other disciplines ["*These units may correspond to a physical hardware device or may refer to assemblies of units for which composite failure modes are identified. The units have been referred to in literature by various nomenclature including systems and subsystems, assemblies and subassemblies, components and subcomponents, structures and substructures, etc.*" (column 6, lines 56-63)].

Cristian teaches the errors recited in the claim as known in the art:

"warm recoverable errors" comprise application failures that can be corrected by a restart without loss of state of the application ["*If, after a first omission to produce output, a server omits to produce output to subsequent inputs until its restart, the server is said to suffer a* crash

failure"; and "*A pause-crash occurs when a server restarts in the state it had before the crash,*" (page 58, right column, emphasis added)], and

"non-warm recoverable errors" comprise application failures that can be corrected by a restart with loss of state in the application ["*If, after a first omission to produce output, a server omits to produce output to subsequent inputs until its restart, the server is said to suffer a* crash *failure*"; and "*An amnesia-crash occurs when the server restarts in a predefined initial state that does not depend on the inputs seen before the crash,* " (page 58, right column, emphasis added)].

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the errors taught by Cristian with the reliability model generation system and method of McMann in order to better analyze and understand a complex system, such as a distributed computing system contemplated by Cristian. Such a system would comprise a model of the network defining the distributed computing system. The combination could be achieved by modeling the distributed system using the method taught by McMann, where the various subcomponents correspond to individual computer systems and the hardware and software on those systems.

Applicants' arguments in favor of claims 8 and 18 refer to McMann's and Cristian's alleged deficiency regarded software components, which has been addressed above.

Claim 18 recites a computer program product comprising computer readable code for performing the method of claim 8. As McMann is a computer-implemented method (Fig. 2),

claim 18 is rejected for the same reasons and with the same combination formed in the rejection of claim 8.

7. Claims 9-12 are rejected under 35 U.S.C. § 103(a) as being unpatentable over McMann in view of Cristian, and further in view of Malek.

Neither Cristian nor McMann explicitly teach determining a fraction of recovery failures for warm or non-warm recoverable errors as recited in the claim. However, as noted in the rejection of claim 8, Cristian does teach "warm recoverable errors" and "non-warm recoverable errors".

Malek teaches contributing factors to the mean time to repair (MTTR) which are functionally equivalent to a fraction of recovery failures, such as T_B, "time required to run diagnostics or analyze information logged at the time of the error to determine the fault symptoms"; P_{diag}, "probability that the diagnostics or logout analysis will be effective in determining the fault symptoms"; and T_E, "time required to run the diagnostics to verify that the problem has been fixed" (page 233, left and right columns). Malek teaches MTTR from the perspective that the fault will be eventually corrected; as such, a "recovery failure" is represented by the inverse of P_{diag}, that a fault will be incorrectly diagnosed and time treating it, $T_D(i)$ and T_E, will be lost. Malek considers the probability of misdiagnosis of the fault (a probability being a number between 0 and 1), which leads directly to a failure to recover.

It would have been obvious to a person of ordinary skill in the art to combine the MTTR model taught by Malek with the combined reliability model generation system and method of

McMann in view of Cristian in order to more accurately model the state transitions from failure to operational, especially when concerned with simultaneous failures (McMann, column 5, lines 58-63). The combination could be achieved by using Malek's MTTR model when computing the state transitions for a recovery action.

Regarding claims 13-15, McMann teaches that the reliability model includes a model for defining parameters of the node, such as expected failure rates and time to recover from the expected failures for components of the platform ["*Given the state space description, including an identification of the initial state and those states that represent an unreliable system, SURE computes the* *__upper and lower bounds on system reliability__* *and provides an enumeration of all system failures,*" (column 6, lines 8-12, emphasis added); transitions include recovery actions, as in "*The failure modes and* *__FDIR attributes__* *are described to ASSIST as* *__transitions in the form of logical statements,__*" (column 6, lines 20-21, emphasis added); failures and recoveries, represented by transitions, are time and rate dependent, as in "*Another critical aspect of FMEA is concerned with the effects of multiple failures on the system and the effects of nearly simultaneous failures – a particular state of vulnerability in which a second failure may occur before the system can recover from the first failure. These time dependencies contribute to the difficulty of an accurate reliability analysis*", (column 5, lines 58-64)]. Failure rates and recovery rates are used to generate state transition parameters ("*State transitions, in SURE, describe the occurrence of faults and fault recovery actions that cause the system to change from*

one state to another", (column 6, lines 5-8)]. McMann explicitly teaches considering the recovery times for subcomponents and components.

Official Notice is taken that in the case of a network or distributed computing system, it is known in the art that subcomponent (node) reboot time and component (network) reboot time significantly contributes to the recovery time of the system and should therefore be contemplated as parameters of the corresponding recovery actions.

8. Claims 16 and 19 are rejected under 35 U.S.C. § 103(a) as being unpatentable over McMann in view of Malek.

McMann teaches a system and method for generating a reliability model for a complex system having different classes of failures ["*The present invention provides a reliability model for use by a reliability analysis tool.*" (column 5, lines 37-54); "*A system in SURE is defined as a state space description: the set of all feasible states of the system, given an initial state. State transitions, in SURE, describe the* <u>*occurrence of faults and fault recovery actions*</u> *that cause the system to change from one state to another,*" (column 6, lines 3-8, emphasis added); "*For highly reliable system, additional functions are incorporated into the architecture for* <u>*failure detection, isolation, and recovery*</u> *(FDIR),*" (column 5, lines 54-56, emphasis added)];

The reliability model defines a recoverable state for a modeled error ["*State transitions, in SURE, describe the occurrence of faults and* <u>*fault recovery actions that cause the system to change from one state to another*</u>*. Given the state space description, including an identification of the initial state and those* <u>*states that represent an unreliable system,*</u> *SURE computes the*

upper and lower bounds on system reliability and provides an enumeration of all system failures," (column 6, lines 5-12, emphasis added)]; and

The reliability model determines a failure rate for said error and a recovery rate for said error [See above, also failures and recoveries, represented by transitions, are time and rate dependent, as in "*Another critical aspect of FMEA is concerned with the effects of multiple failures on the system and the effects of nearly simultaneous failures – a particular state of vulnerability in which a second failure may occur before the system can recover from the first failure. These time dependencies contribute to the difficulty of an accurate reliability analysis",* (column 5, lines 58-64)].

McMann does not explicitly teach determining a fraction of recovery failures for warm or non-warm recoverable errors as recited in the claim.

Malek teaches contributing factors to the mean time to repair (MTTR) which are functionally equivalent to a fraction of recovery failures, such as T_B, "time required to run diagnostics or analyze information logged at the time of the error to determine the fault symptoms"; P_{diag}, "probability that the diagnostics or logout analysis will be effective in determining the fault symptoms"; and T_E, "time required to run the diagnostics to verify that the problem has been fixed" (page 233, left and right columns). Malek teaches MTTR from the perspective that the fault will be eventually corrected; as such, a "recovery failure" is represented by the inverse of P_{diag}, that a fault will be incorrectly diagnosed and time treating it, $T_D(i)$ and T_E, will be lost. Malek considers the probability of misdiagnosis of the fault (a probability being a number between 0 and 1), which leads directly to a failure to recover.

It would have been obvious to a person of ordinary skill in the art to combine the MTTR model taught by Malek with the combined reliability model generation system and method of McMann in view of Cristian in order to more accurately model the state transitions from failure to operational, especially when concerned with simultaneous failures (McMann, column 5, lines 58-64). The combination could be achieved by using Malek's MTTR model when computing the state transitions for a recovery action.

In response, Applicants argue primarily that:

> The Malek MTTR does not teach a fraction of "failures" from recovery, i.e., a fraction of times that recovery is not successful. Hence, nowhere in Malek is a value of "a number of failures to recover from said error" discussed, and such a value is not part of the MTTR. The Examiner construes "Pdiag" as teaching this concept as this variable is addressing the probability the diagnostics will be effective, but Applicants could find no indication that such a probability is calculated in the same fashion as the fraction of recovery is defined in claim 16.

The Examiner respectfully traverses this argument as follows:

The Examiner respectfully submits that the claim language broadly requires "determining a fraction of recovery failures". A complete lack of recovery failures denotes a fraction of zero. A person of ordinary skill in the art would recognize the equivalence of "dividing a number of failures to recover from said error by a number of attempted recoveries from said error" and merely assigning the value to be zero in a model that does not include recovery failures. Assuming Applicants' argument is persuasive, Malek would still teach the limitation related to recovery failures because the language does not require a non-zero fraction.

However, the Examiner respectfully submits that Malek teaches the limitation regardless of the interpretation of "a fraction of recovery failures". The process of diagnostics in response to a failure would be recognized by a person of ordinary skill in the art as a recovery action.

Malek expressly teaches a probability that the diagnostics are successful, the inverse being the probability that the diagnostics, a recovery action, fails. Further, it would have been obvious to a person of ordinary skill in the art that the probability of an event is often defined as (number of successes) divided by (number of attempts), which is recited by the claim.

Applicants' arguments have been fully considered, but have been found unpersuasive.

Claim 19 recites a computer program product comprising computer readable code for performing the method of claim 16. As McMann is a computer-implemented method (Fig. 2), claim 19 is rejected for the same reasons and with the same combination formed in the rejection of claim 16.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Leo Picard can be reached at (571) 272-3749. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Proctor
Examiner
Art Unit 2123

jsp

Paul L. Rodriguez 9/28/05

Primary Ex
Art Unit